ARMY RESEARCH LABORATORY

Software for Real-Time Analysis of Subsonic Test Shot Accuracy

by Sam Spangler

ARL-TR-6880 **March 2014**

NOTICES

Disclaimers

Army Research Laboratory
Aberdeen Proving Ground, MD 21005

ARL-TR-6880 **March 2014**

Software for Real-Time Analysis of Subsonic Test Shot Accuracy

Sam Spangler
Human Research and Engineering Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. **REPORT DATE** *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED (From - To)**
March 2014	Final	June to August 2013

4. TITLE AND SUBTITLE
Software for Real-Time Analysis of Subsonic Test Shot Accuracy

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Sam Spangler

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: RDRL-HRS-B
Aberdeen Proving Ground MD 21005

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-6880

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
William.h.harper52.civ@mail.mil

14. ABSTRACT
When testing weapons that fire subsonic ammunition, analyzing the accuracy of gunshots fired at paper targets can be very tedious. The U.S. Army Research Laboratory (ARL), Human Research and Engineering Directorate (HRED), Dismounted Warrior Branch (DWB) is working on making the analysis of subsonic shot accuracy faster, less tedious, and more efficient. My Science and Engineering Apprenticeship Program (SEAP) project was to aid in the design and programming of an application that digitally analyzes a target and provides feedback on the location of shots fired in real time. I used the C++ programming language, the Open Source Computer Vision (OpenCV®) software library, and Microsoft Windows® Application Programming Interfaces (APIs) to create the application. The software used a microphone to detect when a shot was fired, and a webcam to capture frames of test fire video for comparison through OpenCV image analysis tools. Based on the comparison, the software then computed the coordinates of each shot relative to the center of the target before plotting location points on the screen for real-time accuracy feedback to the shooter. I conducted two software validation studies using the software at an HRED test fire research facility, the results of which are described in detail within this report.

15. SUBJECT TERMS.
Shooting performance, shot accuracy, video scoring, subsonic

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					William H. Harper
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** (*Include area code*)
Unclassified	Unclassified	Unclassified	UU	16	(410) 278-5966

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

Acknowledgments

First and foremost, I would like to thank Dr. Ellen Haas for all her support and encouragement throughout the past two years I've spent as a Science and Engineering Apprenticeship Program (SEAP) student. She has been a great mentor and has become someone I look up to. I would also like to thank Mr. Bill Harper for his guidance over these past two years and Mr. Alexander Miller for helping me with coding and project guidance along the way. In addition, I would like to thank the entire U.S. Army Research Laboratory (ARL), Human Research and Engineering Directorate (HRED), Dismounted Warrior Branch for creating a great working environment and making work fun. Finally, I want to thank the SEAP. It has been an awesome two years of fun and learning.

Student Biography

This year I returned to the U.S. Army Research Laboratory (ARL), Human Research and Engineering Directorate (HRED) for my second year as a Science and Engineering Apprenticeship Program (SEAP) student. My first year was spent creating tablet-based questionnaire applications for easier, faster ways for researchers to receive feedback from participants in field studies and laboratory experiments. This past June I graduated from Aberdeen High School, and come this fall, I will be a freshman Retriever attending the University of Maryland Baltimore County (UMBC). I plan on following in my mother's footsteps and majoring in Mechanical Engineering while attending UMBC. My future plans include graduating from college and getting a job as an engineer at Aberdeen Proving Ground, MD.

1. Background

With current instrumentation, it can be difficult for researchers to analyze the accuracy of weapons that fire subsonic ammunition. Acoustic sensors placed on a firing range can detect specific shockwaves of supersonic projectiles passing through the air, allowing the coordinates of the shot to be calculated. However, acoustic sensor systems cannot detect shots made by a weapon firing subsonic ammunition. In addition, detection of supersonic projectiles can be inaccurate when shot at greater ranges. Thus, when testing Soldier weapons that fire subsonic ammunition, U.S. Army Research Laboratory (ARL), Human Research and Engineering Directorate (HRED) researchers are left to analyze the accuracy of subsonic shots by hand, using basic measuring tools (stop watches and manual measurements of shot placement), which are methods very much subject to human error. Other facilities employ post-processing analysis software with digital pictures of targets. With many new handgun studies anticipated in the near future, Dismounted Warrior Branch (DWB) researchers needed a faster, less tedious, and more efficient method of analyzing firing accuracy of subsonic ammunition weapons. As small-arms technology advances, the need for the ability to collect meaningful data metrics, such as location of hit, as well as the need to reduce human error and undue burden, has necessitated a corresponding advancement in data collection technology and methodology.

2. Project Objective

The purpose of my Science and Engineering Apprenticeship Program (SEAP) project was to aid with the design and programming of an application that analyzes and reports real time feedback on the location of shots fired on a target. This software was expected to reduce time and resources spent on manually analyzing shots fired, thereby increasing efficiency of weapons testing. This software can provide DWB, as well as the larger Department of Defense (DOD) community, with a much needed manner in which to achieve quality assessments and evaluation of Soldier-system performance with small arms.

3. Method

3.1 Software

With a goal of running the shot analysis software in parallel with other C++ coded software, DWB researchers chose to use the C++ programming language because it could easily implement external software libraries. I began my project by learning how to use C++ functions and commands. Although I had never worked with C++, I found it quite easy to learn due to significant similarities to the Java programming language, which I had previously learned.

DWB researchers wanted to use the Open Source Computer Vision (OpenCV) software library for capturing and analyzing frames of video. OpenCV contains numerous premade sets of programming functions for real-time image processing and analysis and allows quick implementation of core features of the application. I also learned how to use this software.

Microsoft Windows application programming interfaces (APIs) also contain numerous sets of functions used on Windows platforms. I learned and used these APIs to create the user interface for the application, which consisted of a window with a menu bar, several function buttons and a text field for displaying shot coordinates. Windows APIs were also used for capturing, processing, and analyzing sound input as well.

To bundle all of these components together, I learned and used the Microsoft Visual Studio 2012 Express Edition. This software compiled all the code, pulled all the necessary resources together, and executed all of the called operations.

3.2 Instrumentation

In order to function, the software needed a microphone and video camera to receive audio inputs and record video data. The best options available were the built-in laptop microphone for audio input and a standard high definition webcam for video recording. The software uses this equipment to instantly capture and process both audio and video data, and compare captured frames of video to determine where the bullet hit the target.

3.3 Software Methodology

The software methodology consisted of three steps. The first was to detect the gunshot, the second was to capture and process the image, and the third was to determine the location of the shot on the target.

Step 1: Detect gunshot. Once ready for testing, the microphone started detecting sound while waiting for a gunshot to occur, which was defined by a large spike (increase) in sound input into the microphone. The software placed the microphone input into one of several sound buffers, and analyzed the sound data while another sound buffer was being filled. This process of overwriting and analyzing multiple sound buffers allowed the software to analyze sound input in real time. Buffer overwriting ran in a loop until a gunshot (a sound consisting of 70% of microphone volume input) was detected.

Step 2: Capture images. Upon detecting a gunshot, the software sent a signal to the camera to capture two images. The first image was a frame of video before the projectile made any contact with the target, and the second frame showed the target after the projectile had gone through it. These images were then processed, as described below.

Step 3: Process and compare images to determine the location of the shot on the target. The two images were processed and compared using the methods of grayscaling, image subtraction, and thresholding.

First, with grayscaling, the two images captured from the webcam were converted from color to grayscale images. This image processing was accomplished by averaging all the red, green, and blue (RGB) channel values within each image. These values were stored in a giant matrix, which the computer could interpret and display. By grayscaling the images, the remainder of the image processing was much simpler to complete.

Next, after both images had been converted to grayscale, the software obtained the difference between the two images by subtracting the image matrices. Any pixels within the images that were identical (including shots that have already been fired) canceled each other out and left a black pixel behind. Any pixels that were different were subtracted and changed, as scanning of the image was continued. This step in the process left a subtracted image behind for the computer to work with. The software then applied a binary threshold to the subtracted image, turning any pixel below a certain value completely black and any value above that value completely white. Once the threshold was applied to the entire image, the computer had only black and white pixels to compare. The result was white pixels showing against a black background.

After obtaining a black and white subtracted image, the software then scanned the entire image to try to identify the pixels that represented the bullet hole. The application used an algorithm that found and assessed clusters of white pixels. If clusters were too small or too large, they were discarded from a list of possible bullet holes. Remaining clusters were given point values based on how many of the pixels were close to others. The cluster with the highest point value was finally chosen as the shot. If a cluster was not chosen, the software determined that the shooter missed the target when firing. Once the software identified a bullet location, it placed a point reflecting a bullet hole on the target image of the computer's application interface to show the user the location of the bullet hole, and printed the coordinates of that shot in the interface coordinate box. The application interface is shown in figure 1. Due to the speed of computer processing technology, the entire process of capturing sound and video input, comparing frames and determining the location of the shot, occurred within milliseconds.

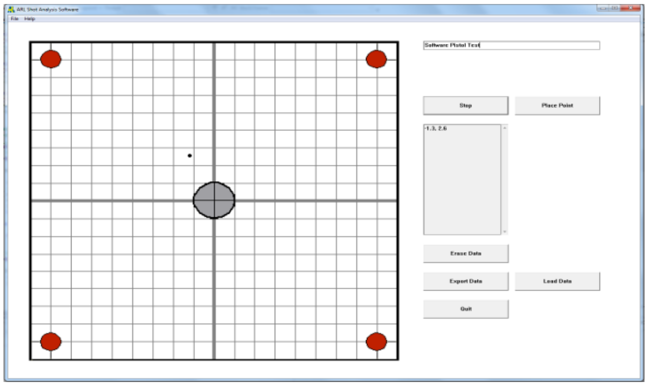

Figure 1. The application interface, which contains picture of target.

4. Research Results

The shot accuracy software was tested twice. The first test was at the HRED M-Range live fire research facility on July 30, 2013. The second test was conducted at the same range facility on August 6, 2013.

4.1 First Test

The main priority of the first test was to check the image capture and processing software. The sound input and analysis software was incomplete, and therefore, could not be tested. Test results indicated that the threshold step in the image processing was dependent on how much sunlight hit the webcam lens, causing the software to be less efficient in bright light conditions. In order to counteract negative effects of bright sunlight, DWB is planning future work to implement a user-controlled manual threshold value selection or possibly create a dynamic threshold algorithm that changes the threshold value at which the image is changed, based on how much of the subtraction image is visible.

Another problem uncovered in the first field test was that algorithm performance in the image processing subtraction step worked less efficiently with the colors that we used as target corner points and outlines. Our initial targets contained red corner points and thick black outlines, which presented a problem because not enough of the lines and corner points were cancelled out during the software subtraction process. To eliminate this problem, I changed the target design to cyan corner points and thin grey outlines. I hypothesized that this would be successful because if the corner points and outlines were lighter, the software had a better chance of recognizing the bullet hole, which is usually dark in color.

4.2 Second Test

For the second test, held a week later at an HRED firing range, our priority was testing the sound input and analysis software. At this time, the sound input and analysis code had been implemented and some interface improvements were made to the application. With a majority of the core components of the application complete, the test results were positive; when the audio software detected a gunshot, the camera quickly captured two frames of video, compared them, and saved them. All of the shots were successfully compared and saved, but 30% of the time, the first images were missed because the camera did not capture a first image quickly enough before the round made contact with the target. DWB researchers and engineers discussed simple changes could be made to the code to speed up image processing in order to be able to capture the first frame more quickly. Figure 2 shows the testing of the software at the HRED firing range (with the user in the foreground).

Figure 2. Testing the application.

Although some coding changes can be made to improve software efficiency, there are still some factors that can alter the performance of the shot analysis software. These factors include the distance between the microphone and the shooter, the distance between the shooter and the target, computer processing capabilities, and other miscellaneous factors. For one miscellaneous factor that we found in the second test, if a shooter fires a shot that travels through a previous bullet hole, the software determines that the shot was a miss. DWB plans to implement a feature that allows the user to manually place a point on the target using the interface software if they believe that the software did not function correctly.

5. Future Work

As described above, many improvements will be made to the application. Once fully developed, the software will be tested by DWB researchers, who will calculate the average error between the new software and previous DWB subsonic shot analysis methods. DWB also plans to refine software accuracy to allow more target images to be captured. In addition, DWB plans to explore the integration of my software with HRED's portable target system, which uses multiple targets that pop up and down at preprogrammed intervals. This would enable DWB researchers to capture shot accuracy on several targets controlled by the portable target system.

6. Conclusion

Current methods of shot tracking and analysis make it tedious and difficult to measure the accuracy of weapons that fire subsonic ammunition. With multiple handgun studies anticipated for the near future, ARL HRED needed a faster and more efficient method to analyze the location of shots fired by Soldiers during weapons testing. To resolve this issue, I designed and programmed a software application to analyze subsonic shots in real time. The software uses a microphone to record audio for detection of the weapon firing and a webcam to record video of the target for analysis. The shot accuracy software uses a specific methodology to define the location of the shot on the target. The feedback is then stored to be exported as text in the software interface. This software is expected to reduce time and resources spent on manually analyzing targeting accuracy, thereby increasing efficiency of DWB Soldier weapons testing. This software could be the start of a great step forward in shot detection technology.

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDFS)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL IMAL HRA MAIL & RECORDS MGMT
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
2 (PDFS)	DIRECTOR US ARMY RESEARCH LAB ATTN RDRL HRS B W HARPER

INTENTIONALLY LEFT BLANK.